International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Computation of Similarity Between Two Pair of Sentence Using Word-Net

Atul Gupta*[1], Kalpana Sharma[2], Krishna Kumar Goyal[3]

Submitted: 23/01/2023 **Revised:** 14/03/2023 **Accepted:** 08/04/2023

Abstract: In the current era the data is available enormously and abundantly, but to find relevant and accurate data from the availability is a humongous task. Searching is required to be accurate and exact then it gives a satisfaction., But path based or edge counting approach, Information content approach, feature based approach and Hybrid approaches unable to provide the satisfactory search result. Current available algorithms are not that efficient to provide exact and accurate search result. In this paper we have implemented and found a better similarity computation compared with existing algorithms. Calculation of similarity between sentence pair based on Word-Net noun IS-A relationship and verb relationship have been done. The proposed algorithm is at par with the mean human similarity measure and it performs efficiently in sentence similarity computation too.

Keywords: *Semantic, Similarity, Verb, WordNet, Stem, Derivation Noun.*

1. Introduction

In the field of Natural Language Processing (NLP), calculation of similarity between word pair and sentence pair is still an open question. Generally, the conceptual distance of two objects plays a vital role on computation of semantic similarity [1], based on their correspondence meaning. Computation of similarity in NLP has various applications. In the application of Bio-medical to analyse gene clustering results and genes prioritization [2-4] semantic similarity tool is used. In the application of internet, semantic similarity is used in search engine queries [5]. There are many more applications of semantic similarity computation such as information retrieval on web [6] and text categorization [7]. Hence, for the above applications the robust methodology is needed to compute the similarity in different type of domains.

The above-mentioned application is specific [2,3] to a particular domain which require a different algorithm although the basic approach on computing the similarity value is same. To identify the close-ness between objects there is a need of some standard predefined measure which describe relatedness of meaning [13]. In the absence of predefined measure there is a problem in measuring similarity. Now at this point, role of lexical databases comes into play. In lexical database there is a connection of words which can be used for measuring similarity value between words [8]. A lot of application has been defined in the last few years which is proved to be efficient on measuring of similarity [9-12].

This paper focused on to develop a improved version of existing algorithm[1,8,10,34,] that is robust and efficient, while integrating with corpus of domain specific. The core objective of this research to construct an algorithm based on semantic similarity which is robust and perform well on comparison with other existing algorithm over Rubenstein and Good-enough[13] standard benchmark dataset. The computation of similarity is done on treating the course objective as the sentence in the NLP and then aligned with statistics of domain specific. In the next section there is a discussion about related work. The complete methodology in step-by-step manner is discussed in section-3. To traverse the lexical database with the working example is discussed in section-4. The proposed similarity calculation for pair of sentence approach in pilot data set [14] is discussed in section-5. The result obtained in proposed approach are compare with existing algorithm is discussed in section-6. Lastly, there is a brief description and conclusion given in section-7.

2. Literature Review

The work on semantic similarity by the researchers on word and sentence is highly significant. The authors in this paper [9] proposed three experiments on semantic similarity. In the short-term memory (STM) investigation, it is observed acoustic similarity had a strong negative effect. But when tested on word sequences, it has got a substantial influence. In the final experiment when the authors use the same concept on visual they got a significant effect. Information-theoretic definition of similarity that can be used with a

[1] *Bhagwant University, Ajmer, Rajastan , India*
[2] *Bhagwant University, Ajmer, Rajastan , India*
[3] *Raja Balwant Singh Management Technical Campus, Agra, India*
** Corresponding Author Email: atulcdac25@gmail.com*

probabilistic model is presented in[1] and applied to different areas like biomedical, gene prioritization, etc. Based on shared information content [10] the authors proposed a IS-A taxonomy semantic similarity method. The proposed algorithm is implemented for resolving the syntactic and semantic ambiguity. The use of ontological annotation for evaluating the knowledge content similarity between items in dataset is investigated by the authors[3]. A suggested Information Retrieval model for retrieving the documents from the web is presented in [5]. The computation of similarity between sentence pairs was measured through corpus statistics on the lexical database is presented in[12]. In the biomedical domain, the authors discussed six domain-independent measures in [4]. They also developed and proposed a context vector measure based on medical corpora which can be utilized as a semantic relatedness indicator. With the help of a corpus-based measure of semantic word similarity and the modified version of the Longest Common Subsumer (LCS) algorithm. The authors tried to present a method for determining the semantic similarity of texts in [28]. The proposed method can be used in a range of textual knowledge representation and knowledge discovery applications. By using the PageRank, graph-centrality algorithm [29] the author demonstrated by incorporating the measure of the relevance of sentential words. It was a statistically significant improvement in clustering performance using a variety of external factors. Regarding a problem on semantic similarity between queries in a question answering system is considered and proposed an approach on domain-specific taxonomy with three factors named as Structure of Taxonomy, Mapping of Keyword, and Weight of Keyword in [30]. A finite-state approach is presented in [31] for finding the semantic similarity of two sentences. The proposed method has been designed by using the concept of bi-directional logic as well as a semantic ordering technique. Classification of sentence similarity has been done in [32]. By using the grammar and semantic corpus-based similarity algorithm, an algorithm was proposed in[33] for sentence similarity computation. On the syntactic structure in[34], proposed a sentence similarity algorithm that first converts the sentence into words, based on syntactic structure, then converts word similarity into concept similarity through word disambiguation, and finally computes the sentence similarity. The computation of Semantic Textual Similarity between words, sentences, paragraphs, and documents is classified and presented in[35] for feature reference. A supervised regression-based model is presented in [36] that effectively incorporates the various similarity computation metrics. A comparison-based sentence similarity is proposed in [37] based on three components i.e. Lexical similarity (Lexsim), Semantic similarity (SemSim), Syntactical Semantic Similarity (SynSemSim). Computation of similarity between word-pairs and sentence-pairs is done by edge counting method on a lexical database. With the help of the edge counting mechanism on the lexical database, the proposed algorithm in [38] is computing the similarity between word pairs and sentence pairs. In[39], a semantic similarity algorithm is the improved version of the path-based method and the most recent state-of-the-art method (Word2Vec) for semantic similarity computation. The proposed algorithm has tested a dataset of 162 Bangla word pairings that had been annotated by five experts. In [40], a similarity computation that has been done based on word-based, structure-based, and vectored-based. A classification review-based presentation on Traditional Natural Language Processing (TNLP) is presented in[41]. The authors classify the TNLP as knowledge-based, corpus-based, deep neural network-based methods, and hybrid methods based on their underlying principles. For semantic similarity testing purposes, a new dataset named CoSimLex which contains pairs of words was constructed by the authors in [42]. For determining the semantic similarity, the author in [43] uses the Discourse Representation Structure (DRS) of natural language phrases. The experimental results show that the use of structural information produces better outcomes than simple word-to-word translation. In [44], the term similarity defined in three ways i.e. term-similarity, sentence-similarity, and document-similarity. Two semantic models i.e. knowledge-based and prediction-based also discussed by the authors. Though different semantic methods exist, there is still a requirement of the semantic algorithm which boosts the overall result of the studied algorithm. As the importance of semantic similarity increases, most of the existing algorithms are fail to provide accuracy in the case of word and sentence similarity.

3. Related Work

Previously, various research have been done in the field of NLP. In this paper the valuable contribution is given in computation of similarity between word pairs and sentence pairs. In this section some review have done to identify the advantages and disadvantages of previous methodologies, and identify some level difficulties on computation of similarity between pair of words and sentences. The computation of similarity can be classify into three different categories:

a. Word Co-occurrence Methodology

b. Lexical database Methodology

c. Search Engine based results Methodology

3.1. Word Co-occurrence Methodology:

The co-occurrence methodology is one of the commonly used method in the area of information retrieval. In this method the words, list of meaningful words and for each and every queries is considered as document itself. Here in this, there is a formation of vector for a query and documents.

The information retrieved is based on query vector and document vector [6].

Drawbacks: Some drawbacks of this method are:

a. The ignorance of word order in the sentence.

b. The word meaning is not considered in the context of sentence.

Advantages:

The matching of documents is done whatever the size of the document. Keyword successfully extracts from the document

3.2. Lexical Database: Methodology

In the pre-defined hierarchal structure of Word-Net there is a huge collection of words, meaning and relation with other words. In the WordNet hierarchy concepts are arranged in a tree structure [12].On computation of word similarity ,path level distance between word pair and depth of sub-sumer in the hierarchy of Word-Net plays a vital role. The sub-sumer is a common node between two words on comparison of similarity between them. The Information-content (IC) value of each and every word in the Word-Net hierarchy also influence the similarity value in final similarity calculation.

Limitation:

Computation of similarity between pair of sentences, the best matching word pair is taken into consideration, if the word meaning was different in two sentences, so the appropriate meaning is not considered.

The IC value of each word vary from corpus to corpus, so the final similarity value of each corpus is different.

3.3. Web Search Engine Result Methodology

Computation of similarity is achieved by search engine results [16]. In this method the word with opposite meaning are comes together in a web page which influence the final similarity score. The method is analysed by Google Similarity Distance [17] and search engine used in the study are go-ogle and Bing. Here, achieved results are not very efficient.

All the above similarity method while computing the similarity does not consider the contextual meaning of the word in the sentences. The proposed work trying to resolve the issues, on disambiguating the word in the sentences and creating semantic vector of words dynamically on comparing words and sentences.

4. The Proposed Approach

In the proposed work consider there is a text which contain a words in sequence on dealing with every words in the sentence discretely, by take care of syntactic and semantic structure of the word. In a lexicographical database, corpora (like Word-Net) the information content value (IC) is directly proportional to the frequency of the word. The calculation of similarity value between sentence pair is divided into three parts: (a) Similarity in words (b) Similarity in sentence (c) Similarity in word-order

In the previously existing approach, on computation of similarity uses vocabulary of fixed structure, but in the proposed methodology a lexical database or corpus like Word-Net is used for the closest meaning word. In this methodology, for each occurrence of word in the sentence vector is created so that there is a weight assigned for each word on comparison with other word in the sentence. Calculation of Similarity value between sentence pair is depend on two semantic vectors. Syntactic similarity is considered between pair of sentences ,order vector is formed. Now, the final value of similarity is based on order vector and semantic vector. The brief description of the steps are described below:

4.1. Similarity in words

The methodology proposed here uses a database that is lexically connected for English words i.e. Word-Net[11] developed by Princeton university. The following steps are involved for computation of similarity:

4.1.1. Identification of words for comparison

Estimation of similarity between words, the identification of words is required for comparison. Here usage of word-tokenization and part of speech (POS) tagging methodology that was implement in NLP toolkit, NLTK [18]. The steps on filtering the sentence i.e. input, it tagged the words of the sentence in different POS and labelled it accordingly. The discussion about Word-Net in section 2 it is well known that there is path relationship between noun and verb only. In other part-of-speeches in Word-Net path level relationship are absent. Hence, the possible link relationship only for noun-noun and verb-verb. Therefore, to make the algorithm efficient in term of time and space, consideration of only noun and verb relations to compute similarity

4.1.2. Association of words with sense

In Word-Net structure computation of similarity is depend on it synonym relationship. Every word present in the Word-Net has synsets , depends upon the meaning in the context of sentence .For example the word ‘bank’ in represent all synonym (synsets) present in the Word-Net. On comparing the distance between synsets, it varies if the meaning of the word changes.

In the example given below the smallest path value distance between word1&word2 i.e ‘river’ and ‘bank’. The word ‘river’ has only one synsets in hierarchy of Word-Net. Estimation of similarity has been done on the basis of path level distance between word- ‘river’ and word-‘bank’ i.e.

consideration of one synsets of word 'river' and three synsets of 'bank' .

The minimum shortest distance between synsets pair of word 'river' and word 'bank' are shown in Table 1. On comparing the similarity between two sentences there are many words those having many synsets. Therefore, if the proper synsets are not considered in context of the input sentences it will reflects error at the early stage of computing the similarity. Hence, word sense plays a vital role on the computation of similarity. In the area of 'word sense disambiguation' research sense or context of the word is identified. The maximum similarity approach, equation-1 is done by Word-sense-disambiguation[19] that is implement in python library pywsd [20]

Table 1. Synset of word river and its shortest path distance in Word-Net

Synset Pair	**Shortest Path Distance**
[river.noun.1] and [bank.noun.1]	8
[river.noun.1] and [bank.noun.9]	10
[river.noun.1] and [bank.noun.6]	11

$$argmax_{syn(a)} \sum_{i}^{n} max_{syn(a)} \left(sim(i,a)\right) \quad (1)$$

Where,

$max_{syn(a)}$ is represented for maximum synset value between $\left(sim(i,a)\right)$

4.1.3. Calculation of Shortest path distance

The methodology which is used to compute shortest path distance is explained with the help of given example: consider words are: $word(w1) = motor - cycle$ and $word(w2) = car$ Now by Word-Net hierarchal structure synset of motorcycle is ('$motor - cycle.n.01$') and synset of car is ('$car.n.01$'). On travelling the path in the Word-net hierarchal structure $motor - cycle \rightarrow moto - vehicle \rightarrow car$. So, the value of shortest path distance between car and motor-cycle is 2. In the Word-Net hierarchal structure similarity between word decrease as the distance between words increases. Here, there is monotonically down function [12] is established to compute similarity between words.

$$f(k) = e^{-ak} \quad (2)$$

Where,

k mean the shortest distance between words

a is constant and the value of a is taken as 0.2 as discussed in [8]

The reason for selecting the exponential function (e) because, f(k) always lies between 0 to 1

4.1.4. Words Distribution in Hierarchal form

The super- sub-ordinate relation is one of the primary relationships present between synsets, it is called as hyponymy, hyperonymy or Is-A relationship [21]. In a Word-Net the relationship between words is from general concept to more specific concept.

The words 'vehicle' has more generalized properties whereas hypnyms relationship of word vehicle have more specialized properties. Therefore, the words that are present in the top layer in the Word-Net have more generalized characteristics and less informative as compared to word present in the lower layer in the Word-Net

There is a vital role of hierarchal distance if the path distances between words are same. In the following pair of words such that 'motor-cycle and car' the path distance is 2.Hence, if the words pair subsume word at the lower- level in hierarchy than for similarity calculation there is need to scale up and if the word are at the upper-level in the hierarchy then there is need to scale down. The established function is given by:

$$G(h) = (e\text{^}bh - e\text{^}(-bh))/(e\text{^}bh + e\text{^}(-bh)) \quad (3)$$

The value of b is 0.45 as discussed in [8].

4.2. Information content value of the word

Various domain has different meaning of the word. This behaviour of NLP (natural language processing) is used for computation of similarity in various domains. The domain of the words always influences the computation of similarity measure. On analysing the information content value of the word, considered word 'bank' . There are two context of the word bank first in Potamology means the study of river and second in Economics means the financial institutions.

On using WSD (Word Sense Disambiguation) which was described in previous section. The Similarity value which was calculated finally is different from corpus to corpus. The corpus which is belong to particular area of domain, it works like a supervised data for algorithm. The actual sense of the word is found by disambiguating the corpus and then compute the frequency of the word.

4.3. Measuring Sentences similarity

The meaning of the sentences is determined by the words in the corresponding sentences Li[12].The use of semantic information in section 3.1 and computation of final similarity value in section 3.2 are discussed. The computation of similarity between sentences discussed previously use static based approach i.e. by using pre-compilation list of words and it phrases. The words retrieve in the precompiled list doesn't retrieve correct information, which is the drawback of this strategy.

The dynamic based approach uses the concept of joint-word vector which compiled different word from sentences. For long sentence, the approach gives inaccurate result.

The approaches find the value of semantic vector for sentences and the goal is to minimize the value of semantic vector. In the method discussed previously, there are various overhead to form value of semantic vector, but dynamic approach avoid this overhead. On the computation of semantic vector all the connectives are eliminate like conjuction, pre-position and interjection. In section 3.1.2 vector size is based on number of retrieved tokens.

Semantic vector are initialized with the positive value and discard all the negative as well as null value. The final value of calculated similarity is based on most of the word which are similar in the sentence.

s1=["A"]["jewel"]["is"]["a"]["precious"]["stone"]["used"]["to"]["decorate"] ["valuable"]["things"]["that"]["you"]["wear"]["such"] ["as"]["rings"] ["or"] ["necklaces"].

s2=["A"]["gem"]["is"]["a"]["jewel"]["or"]["stone"]["that"]["is"]["used"]["in "] ["jewellery"].

The tagged word list length in sentence s1 is 9.

The tagged word list length in sentence s2 is 5.

To reduce the overhead of computing, we eliminate words like a ,as ,or ,you ,to ,is e.t.c. The formation of semantic vector is done by the semantic information about the words that are occur in sentence pair. The semantic vector for sentence1 is shown in the example:

Vector1=[0.99692214, 0.90208685, 0.4220098, 0.0, 0.0, 0.40580856, 0.0, 0.58145326, 0.82854105]

Value of semantic vector information for sentence1 as well as sentence2 is present in vector1. Similarly semantic vector information for sentence2 and sentence1 is present in vector2. Computation of similarity value is done by using two vectors VEC1 and VEC2, the magnitude of the normalization of vector is done by:

$$S = ||VEC1|.|VEC1|| \quad (4)$$

The method produces better result in longer form of sentences by introducing a new parameter lambda (λ), which is evaluated dynamically at the run-time . This method is used in comparison of two paragraph that contain multiple sentences on the introduction of parameter lambda.

4.3.1. Calculation Lambda

Maximum similarity between words is directly proportional to vector magnitude. This property is used by introducing lambda factor on comparison of sentences. The synonymy value of words1 and words2 is 0.8021 as given by Rubinstein in 1965[13]. To take this value as a standard, the cell of vector $VEC1$ and $VEC2$ is generated with the value greater than benchmark value 0.8021. The factor lambda is given by:

$$\lambda = Sum(C1, C2/\gamma) \quad (5)$$

$C1$: $Represent\ the\ total\ valid\ count\ of\ element\ VEC1$

$C2$: $Represent\ the\ total\ valid\ count\ of\ element\ VEC2$

The value of λ is fixed to 1.9 to limit the range of similarity vary between [0,1]. Now similarity is calculated by equation 4 and 5.

$$\text{Similarity} = S/\lambda \quad (6)$$

SENTENCES SIMILARITY ()

1. $S1 \leftarrow disambiguate\ sentence1$ // $S1\ means\ list\ of\ tagged\ token\ in\ sentence1$
2. $S2 \leftarrow disambiguate\ sentence2$ // $S2\ means\ list\ of\ tagged\ token\ in\ sentence2$
3. $Compute\ vector\ length\ vec\_len \leftarrow MAXIMUM((length\ of\ S1, length\ of\ S2))$
4. $VEC1, VEC2 \leftarrow vec\_len()$
5. $VEC1, VEC2 \leftarrow vec\_len(word\_sim(S1, S2))$
6. $\lambda = 0$
7. $while(S1)\ do$
8. $if\ (similarity\ value\ of\ word > similarity\_benchmark\_value)$
9. $C1++$
10. $while(S2)\ do$
11. $if\ (similarity\ value\ of\ word > similarity\_benchmark\_value)$
12. $C2++$
13. $\lambda \leftarrow X/\lambda\ where\ X = Sum(C1, C2)$
14. $S \leftarrow ||VEC1|.|VEC2||$
15. $if(X == 0)\ then$
16. $\lambda \leftarrow vec\_len/2$
17. $Similarity \leftarrow S/\lambda$

4.4. Similarity based on word order

On computing the semantic similarity between pair of sentence, semantic nature of the sentence and syntactic structure are considered. Comparison of word indices in both the sentence is done in word order similarity. While computing the similarity based on words and on the lexical database, grammar in the sentence is not considered. Li[12], in this paper for every word of a sentence a number is

assigned and according to the occurrence and similarity a word order vector is formed.

Value of semantic similarity of word is also consider in word order vector. To compare two sentences if in sentence1 there is a word, that were not present in sentence2, So the value assigned having maximum similarity-value of the word in word-order.

The result produced here not always valid and the error is encountered while calculating the final value of similarity

Sentence1= ["A"] ["quick"] ["brown"] ["dog"] ["jumps"] ["over"] ["the"] ["lazy"] ["fox"].

Sentence2= ["A"] ["quick"] ["brown"] ["fox"] ["jumps"] ["over"] ["the"] ["lazy"] ["dog"].

index. Calculation of similarity based on word-order always produce better result when both the sentence have same set of word in different order, but if the sentence have different set of word, then the word-order is useless. Word order role is very negligible in a different sentences as compared with semantic similarity. Hence, consideration of word order in our approach is an optimal feature. The working of word order similarity is shown in a given example:

The result shows both the sentences 1 and 2 produce the same result, when the edge-based methodology is applied using lexical- database, but in that case the word are appears in shuffle order so scaling down the final similarity value is necessary as they give different meaning.

Computation of similarity between two sentence 1 and 2, start with the formation of dynamic vector $VEC1$ and vector $VEC2$ for sentence1 and sentence2. Vector initialization is explained in section 3.3.

The process is initialized i.e. the sentences which have maximum length. The formation of Vector $VEC1$ for sentence1 are initialized to index value of the word present in the sentence1 with the beginning index is 1. So the vector $VEC1$ for sentence1 is:

$$VEC1 = 1, 2, 3, 4, 5, 6, 7, 8, 9$$

Now the vector $VEC2$ concerned with sentence 1 and 2. Formation of vector $VEC2$ is done by comparing every word from sentence2 with sentence1. The vector is filled in such a way that the word in sentence2 is not present in the sentence1 then the vector $VEC2$index value is filled by the word that are present in sentence2. If the word in sentence2 is present in the sentence1 then the $VEC2$ index value is filled by the word that are present in sentence1.

In the example above, the two word 'fox' present in sentence2. In sentence2 the word fox is present and that is also present in sentence1 at index value 9 i.e. in the vector VEC-2 the entry for fox will be 9.So the vector VEC-2 is formed as:

$$VEC2 = 1, 2, 3, 9, 5, 6, 7, 8, 4$$

The final similarity calculation of word-order is:

$$Word_{sim} = \frac{|VEC1 - VE2|}{|VEC1 * VEC2|} = 0.0665$$

The similarity between sentence-1 and sentence-2 is found to be 0.0665.

List1:

[('gem',Synset('jewel.n.01'))],[('jewel',Synset('jewel.n.01'))],[('stone',Synset('gem.n.02'))], [('used',Synset('use.v.03'))], [('jewellery',Synset('jewelry.n.01'))]

List2:

[('jewel',Synset('jewel.n.01'))],[('stone',Synset('stone.n.02'))],[('used',Synset('use.v.03'))], [('decorate',Synset('decorate.v.01'))],[('valuable',Synset('valuable.a.01'))],[('things',Synset('thing.n.04'))],[('wear',Synset('wear.v.01'))],[('rings',Synset('ring.n.08'))],[('necklaces',Synset('necklace.n.01'))]

5. Implementation

The proposed approach is implemented with the help of Word-Net. The statistical information in the Word-Net is used to compute the IC value of the word. The behaviour with external database is tested by a small corpus. The description of precondition required to implement the proposed approach is described as follows.

5.1. Word-Net as Database

Word-Net is a lexical database which is available as online/offline,Word-Net is a corpus that is available online/offline,which is developed by Princeton university. Word-Net 3.0 version is used in our approach that contain more than 1,17,000 synsets . Synset represent all possible meaning of the word that is used in the sentence. Currently Word-Net has different type of synset structure such as noun, adverb, adjective and verb. The lexicon are independent to each other that means there is no connection such as noun and verb are not linked. Super-subordinate (IS-A / HAS-A) relationship is the main relationship present in the Word-Net. As we move up in the hierarchy there are more general relationship. The Word-Net hierarchy has a common root node in all the noun hierarchy is called Entity. Like Word-Net noun hierarchy, verb is also arranged in hierarchal form.

5.1.1. Shortest path distance in Word-Net

In the Word-Net hierarchy, there are four type of subnets such as nouns, adverb, verbs and adjectives. The connection is available in same part of speeches that means computation of similarity is not possible in cross domain.

In the Word-Net hierarchy, evaluation of shortest distance between two word is done by tree structure. Shortest path distance is computed to climb up in the hierarchy of Word-Net and find the meeting point in both the synsets , which is also the synset. The meeting point synset is called as subsumer. The shortest path is computed by counting the number of hops from one synset to another. Consideration of sub-sumer position on two synset plays a vital role on calculation of hierarchal distance. Word-Net hyperonymy (IS-A) relation for both the synset is used in finding sub-

sumer of synset. The approach, move up in the hierarchy of Word-Net until a synset is found, which is common in both the synset. This common synset is called subsumer. The hypernym set is individually find for both synset and their intersection set is subsumer synset. The shortest path synset is taken into consideration if the the synset set contain more than one synset..

5.1.2. Information content (IC) value of the word

The generalized statistical information present in the Word-Net is used to compute IC value of the word. In the hierarchy of Word-Net the frequency of each synset is present. The implementation in section 3.2 used the frequency distribution for computation of IC value of the word.The detailed step by step explanation for sentence similarity is explained in this section.

The tagged word of sentence1(that is sentence s2 in section 3.3) are jewel, gem, use and jewellery and the tagged word of sentence2 (that is sentence s1 in section 3.3) are jewel, stone, use, decorate, valuable, things, wear, ring and necklace. The Synset of this tagged word is used to compute similarity between sentences.

To Compute the similarity between sentences, identify the Synset and find the Shortest-path-distance in all the synset and pick the matching result which is best to form semantic vector. The intermediate list List1 and List2 is formed that contain word and identified synset.

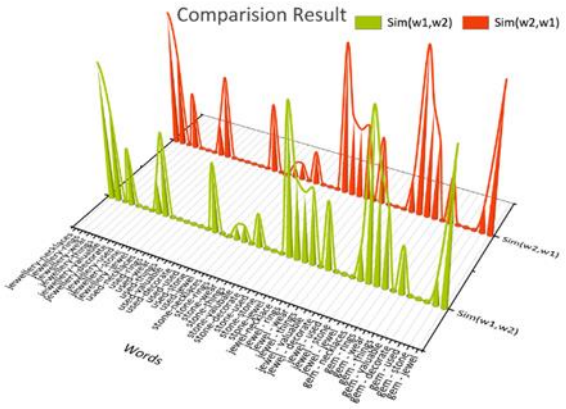

Fig 1: Comparison between word 1 Vs word 2 in List 1 and List 2

The formation of semantic vector for sentence1 and sentence2 is done by comparing Synset of List1 with every synset of List2 is shown in Fig 1. Now the next step is to determine semantic vector size and initialize it to null. According to the method explain in above section 3.3 , the size of semantic vector is 9. In the following part contain cross-compilation of List1 and List2.

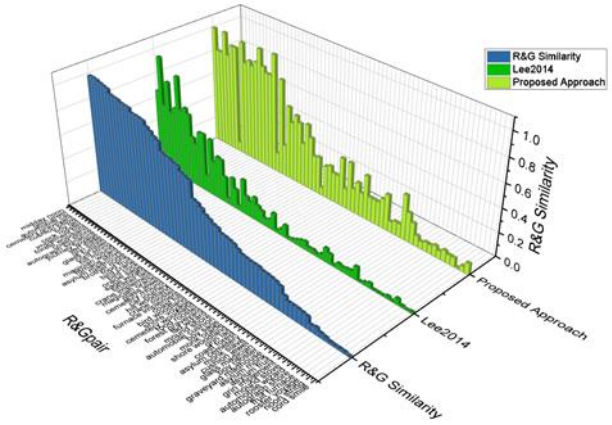

Fig 2: R&G Similarity Vs Lee2014 Similarity Vs Proposed Algorithm Similarity

In fig-2, Our approach is found to be better than Lee2014 but it is closed to the similarity value of R&G Similarity value.

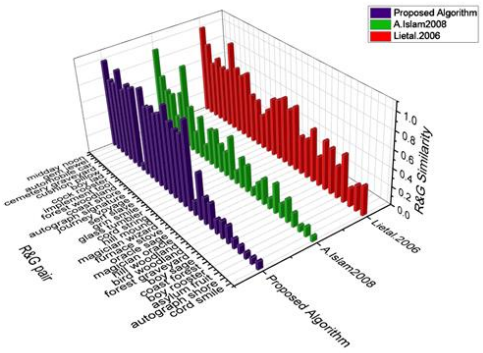

Fig 3: Proposed Algorithm Similarity Vs Islam2008 Vs Li2006

In fig-3, proposed algorithm is compared with Islam2008 and Li2006 and it is observed that our proposed algorithm is bettered that Islam2008 and Li2006. it is found when we have applied a filtering technique to find the suitable similarity between the sentences.

6. Experimental Result and Discussion

The standard set of data that contain 65 noun-pair originally measured by R&G [13], is used in the algorithm. The data set is one of the stable sources to compute semantic similarity and used in over the years. This computation of semantic similarity of word obtained is assisted by standard sentence benchmark dataset by James O' shea [14]. The aim of our approach is to achieved a good result which is closed to benchmark dataset of R&G [13], Collins Co-build dictionary is used for definition of word. The correlation coefficient achieved by our algorithm is 0.875369 which is greater than the previously existing approach. The result of 65 pair of data-set (sentence s1 to s65) compare against benchmark dataset of R&G [13] is shown in Fig 4.

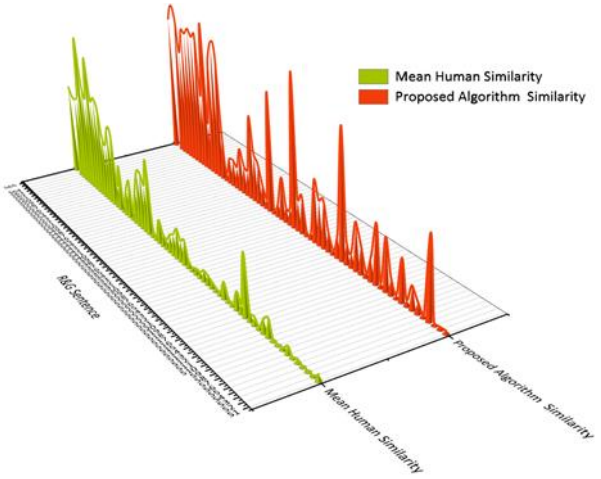

Fig 4: Comparison of Sentence Similarity between proposed Approach Vs Mean Human Similarity

6.1. Sentence Similarity

In Fig: 4 the similarity value of mean human sentence is computed from benchmark dataset of James O'shea [14]. In the paper of Li[12], the explanation is given , that on conducting the survey of 32 participant to compute semantic similarity , marked the sentence not words.

Our Approach outperform with previously implemented methodologies. The better correlation coefficient of 0.879 is achieved by our algorithm. The other approach given by Li[12] and Islam[22] obtained correlation coefficient of 0.817 and 0.854. According to definition present in Collins Cobuild dictionary [23] , 5 sentence pair of words were eliminate out of 65 pair of sentence. The explanation and result are discussed below.

A better pearson correlation coefficient of 0.875 is achieved by our algorithm on similarity measure with R&G [13] word-pair. The performance of our algorithm is better than all the previously implemented algorithm. In Fig 2, proposed methodology of sentence similarity is compared with Lee[24] on taking R&G word pair is shown.

In Fig 3, the comparison of our algorithm with Islam[22] and Li[14] for 30 noun pair and the better performance is achieved. For sentence similarity, there are some words not considered for similarity calculation. The word-pair have more than one common synonym word 17: [coast]-[forest] , 24:[lad]-[Wizard], 30:[Coast]-[hill].For example, In the definition of lad. A lad is a young man or boy and the definition of wizard is the man with magic power. Hence, there are more closely related words pair [man]-[man], [boy]-[man] and [lad]-[man] are present in both the sentences

7. Conclusion & Future Work

In this paper, semantic similarity is computed between two words (word1 and word2), two sentences (sentence1 and sentence2) and two paragraphs (paragraph1 and paragraph2). In the initial phase of the algorithm, first disambiguate the sentence1 and sentence2 and tag them in different part of speech. Disambiguate of words ensure the correct meaning of the word, which is required for comparison. The edge-based methodology which was discussed previously is used in computation of word-order similarity. The IC value of the corpus is used on computation of similarity in particular domain. Semantic vector is used for computing similarity between two words which are formed from sentence and used for computation of sentence similarity. The syntactic structure of the sentence is also considered on word-order vector. The methodology implemented in this paper is compared to the previously used word-pair that contain standard result and human rating result. The pearson correlation for similarity between words is 0.875 and for similarity between sentence is 0. 879.The correlation coefficient achieved by the algorithm is better than previously implemented approach. In future perspective domain of the algorithm will be extended, analyzed the ontologies and relationship in it

8. References and Footnotes

8.1. References

[1] Lin, Dekang. "An information-theoretic definition of similarity." In *Icml*, vol. 98, no. 1998, pp. 296-304. 1998.

[2] Pesquita, Catia, Daniel Faria, Andre O. Falcao, Phillip Lord, and Francisco M. Couto. "Semantic similarity in biomedical ontologies." *PLoS computational biology* 5, no. 7 (2009): e1000443.

[3] Lord, Phillip W., Robert D. Stevens, Andy Brass, and Carole A. Goble. "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation." *Bioinformatics* 19, no. 10 (2003): 1275-1283.

[4] Pedersen, Ted, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G. Chute. "Measures of semantic similarity and relatedness in the biomedical domain." *Journal of biomedical informatics* 40, no. 3 (2007): 288-299.

[5] Freitas, André, Joao Gabriel Oliveira, Seán O'Riain, Edward Curry, and João Carlos Pereira Da Silva. "Querying linked data using semantic relatedness: a vocabulary independent approach." In *International Conference on Application of Natural Language to Information Systems*, pp. 40-51. Springer, Berlin, Heidelberg, 2011.

[6] Varelas, Giannis, Epimenidis Voutsakis, Paraskevi Raftopoulou, Euripides GM Petrakis, and Evangelos E. Milios. "Semantic similarity methods in wordnet

and their application to information retrieval on the web." In *Proceedings of the 7th annual ACM international workshop on Web information and data management*, pp. 10-16. 2005.

[7] Ko, Youngjoong, Jinwoo Park, and Jungyun Seo. "Improving text categorization using the importance of sentences." *Information processing & management* 40, no. 1 (2004): 65-79.

[8] Fellbaum, C. "WordNet: Wiley online library." *The Encyclopedia of Applied Linguistics* 7 (1998).

[9] Baddeley, Alan D. "Short-term memory for word sequences as a function of acoustic, semantic and formal similarity." *Quarterly journal of experimental psychology* 18, no. 4 (1966): 362-365.

[10] Resnik, Philip. "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language." *Journal of artificial intelligence research* 11 (1999): 95-130.

[11] Miller, George A., and Walter G. Charles. "Contextual correlates of semantic similarity." *Language and cognitive processes* 6, no. 1 (1991): 1-28.

[12] Li, Yuhua, David McLean, Zuhair A. Bandar, James D. O'shea, and Keeley Crockett. "Sentence similarity based on semantic nets and corpus statistics." *IEEE transactions on knowledge and data engineering* 18, no. 8 (2006): 1138-1150.

[13] Rubenstein, Herbert, and John B. Goodenough. "Contextual correlates of synonymy." *Communications of the ACM* 8, no. 10 (1965): 627-633.

[14] O'Shea, James, Zuhair Bandar, Keeley Crockett, and David McLean. "Pilot short text semantic similarity benchmark data set: Full listing and description." *Computing* (2008).

[15] Boyce, Bert R., Bert R. Boyce, Charles T. Meadow, Donald H. Kraft, Donald H. Kraft, and Charles T. Meadow. *Text information retrieval systems*. Elsevier, 2017.

[16] Bollegala, Danushka, Yutaka Matsuo, and Mitsuru Ishizuka. "Measuring semantic similarity between words using web search engines." *www* 7, no. 2007 (2007): 757-766.

[17] Cilibrasi, Rudi L., and Paul MB Vitanyi. "The google similarity distance." *IEEE Transactions on knowledge and data engineering* 19, no. 3 (2007): 370-383.

[18] Bird, Steven. "NLTK: the natural language toolkit." In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pp. 69-72. 2006.

[19] Pawar, Atish, and Vijay Mago. "Calculating the similarity between words and sentences using a lexical database and corpus statistics." *arXiv preprint arXiv:1802.05667* (2018).

[20] L. Tan, "Pywsd: Python implementations of word sense disambiguation (wsd) technologies [software]," https://github.com/alvations/pywsd, 2014.

[21] Miller, George A. "WordNet: a lexical database for English." *Communications of the ACM* 38, no. 11 (1995): 39-41.

[22] Islam, Aminul, and Diana Inkpen. "Semantic text similarity using corpus-based word similarity and string similarity." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 2, no. 2 (2008): 1-25.

[23] Dunbar, George. "Looking Up: An Account of the COBUILD Project in Lexical Computing and the Development of the Collins COBUILD English Language Dictionary." (1988): 263-266.

[24] Lee, Ming Che, Jia Wei Chang, and Tung Cheng Hsieh. "A grammar-based semantic similarity algorithm for natural language sentences." *The Scientific World Journal* 2014 (2014).

[25] Gupta, Atul, and Dharamveer Kr Yadav. "Semantic similarity measure using information content approach with depth for similarity calculation." (2014).

[26] Gupta, Atul, and Krishan Kumar Goyal. "Classification of Semantic Similarity Technique between Word Pairs using Word Net."

[27] Goyal, Krishan Kumar. "Computation of Verb Similarity." *Design Engineering* (2021): 4127-4140.

[28] Islam, Aminul, and Diana Inkpen. "Semantic text similarity using corpus-based word similarity and string similarity." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 2, no. 2 (2008): 1-25.

[29] Skabar, Andrew, and Khaled Abdalgader. "Improving sentence similarity measurement by incorporating sentential word importance." In *Australasian Joint Conference on Artificial Intelligence*, pp. 466-475. Springer, Berlin, Heidelberg, 2010.

[30] Nadschläger, Stefan, Hilda Kosorus, Andreas Boegl, and Josef Kueng. "Content-based recommendations within a QA system using the hierarchical structure of a domain-specific taxonomy." In *2012 23rd International Workshop on Database and Expert Systems Applications*, pp. 88-92. IEEE, 2012.

[31] Sitaula, Chiranjibi, and Raj Ojha Yadav. "Semantic Sentence Similarity Using Finite State Machine." *Intelligent Information Management* 5, no. 6 (2013): 171.

[32] Gomaa, Wael H., and Aly A. Fahmy. "A survey of text similarity approaches." *international journal of Computer Applications* 68, no. 13 (2013): 13-18.

[33] Lee, Ming Che, Jia Wei Chang, and Tung Cheng Hsieh. "A grammar-based semantic similarity algorithm for natural language sentences." *The Scientific World Journal* 2014 (2014).

[34] Li, Xiao, and Qingsheng Li. "Calculation of sentence semantic similarity based on syntactic structure." *Mathematical Problems in Engineering* 2015 (2015).

[35] Majumder, Goutam, Partha Pakray, Alexander Gelbukh, and David Pinto. "Semantic textual similarity methods, tools, and applications: A survey." *Computación y Sistemas* 20, no. 4 (2016): 647-665.

[36] Soğancıoğlu, Gizem, Hakime Öztürk, and Arzucan Özgür. "BIOSSES: a semantic sentence similarity estimation system for the biomedical domain." *Bioinformatics* 33, no. 14 (2017): i49-i58.

[37] Wali, Wafa, Bilel Gargouri, and Abdelmajid Ben Hamadou. "Enhancing the sentence similarity measure by semantic and syntactico-semantic knowledge." *Vietnam Journal of Computer Science* 4, no. 1 (2017): 51-60.

[38] Pawar, Atish, and Vijay Mago. "Calculating the similarity between words and sentences using a lexical database and corpus statistics." *arXiv preprint arXiv:1802.05667* (2018).

[39] Pandit, Rajat, Saptarshi Sengupta, Sudip Kumar Naskar, Niladri Sekhar Dash, and Mohini Mohan Sardar. "Improving Semantic Similarity with Cross-Lingual Resources: A Study in Bangla—A Low Resourced Language." In *Informatics*, vol. 6, no. 2, p. 19. Multidisciplinary Digital Publishing Institute, 2019.

[40] Farouk, Mamdouh. "Measuring sentences similarity: a survey." *arXiv preprint arXiv:1910.03940* (2019).

[41] Chandrasekaran, Dhivya, and Vijay Mago. "Evolution of Semantic Similarity—A Survey." *ACM Computing Surveys (CSUR)* 54, no. 2 (2021): 1-37.

[42] Armendariz, Carlos Santos, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Ivan Vulić, and Mohammad Taher Pilehvar. "SemEval-2020 Task 3: Graded Word Similarity in Context." In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 36-49. 2020.

[43] Farouk, Mamdouh. "Measuring Sentences Similarity Based on Discourse Representation Structure." *Computing and Informatics* 39, no. 3 (2020): 464-480.

[44] Varghese, Nisha, and M. Punithavalli. "Semantic Similarity Analysis on Knowledge Based and Prediction Based Models."

Author contributions

Atul Gupta: Conceptualization, Field study, Methodology, and Implementation.

Kalpana Sharma: Data collection, Writing-Original draft preparation.

Krishan Kumar Goyal: Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.